2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing

Modelling of User Requirements and Behaviors in Computational Grids

Joanna Kołodziej
Department of Mathematics and Computer Science
University of Bielsko-Biała
Bielsko-Biała, Poland
Email: jkolodziej@ath.bielsko.pl

Fatos Xhafa
Department of Languages and Informatics Systems
Technical University of Catalonia
Barcelona, Spain
Email: fatos@lsi.upc.edu

***Abstract*—In traditional distributed computing systems a few user types are found having rather "flat" profiles, mainly due to the same administrative domain the users belong to. This is quite different in Computational Grids (CGs) in which several user types should co-exist and make use of resources according to hierarchical nature and the presence of the multiple administrative domains. One implication of the existence of different hierarchical levels in CGs is that it imposes different access and usage policies on resources. In this paper we firstly highlight the most common Grid users types and their relationships and access scenarios in CGs corresponded to *old* (e.g. performance) and *new* (e.g security) requirements. Then, we identify and analyze new features arising in users' behavior in Grid scheduling, such as dynamic, selfish, cooperative, trustful, symmetric and asymmetric behaviors. We discuss also how computational economy-based approaches, such as market mechanisms, and computational paradigms, such as Neural Networks, can be used to model user requirements and predict users' behaviors in CGs. As a result of this study we have provided a comprehensive analysis of Grid user scenarios than can serve as a basis for application designers in CGs.**

***Keywords*-Computational Grids, Scheduling, User Behavior, Security, Resource Reliability, Game-theoretic Models, Neural Networks.**

I. Introduction

The main aim of Computational Grids (CGs) is to virtually combine geographically distributed IT resources from many different administrative domains into one single customized computational infrastructure that enables users to perform computational tasks or data storage capabilities. The Grid resources usually belong to different owners (institutions, enterprises or individuals) and are managed by different administrators. Resource administrators conform to different sets of rules and configuration directives, and can impose different usage policies on Grid users. Despite of advances reported in Grid computing domain, achieving an efficient management of the whole Grid infrastructure under different user types and behaviors still remains challenging.

In traditional distributed computing systems, there are found usually a few user types. As the users belonged to the same administrative domain, their profiles and requirements were rather "flat" and their access to resources *a priori* taken for granted. It is much different the situation in Computational Grids (CGs), in which several user types should co-exist and perform their tasks under the complex nature of Grid environments. Grid users can have very different requirements implying a variety of users' relationships and behavior scenarios. These include selfishness, cooperativeness, trustworthiness, leadership, etc. It is thus very important to analyze and model such user requirements and behaviors to predict the users' needs and actions in order to optimize the Grid system performance at both individual and global levels.

The objective of this work is to present a comprehensive set of Grid users' scenarios to provide a basis for application designers in CGs. We firstly specify a categorization of the Grid users types and define their relationships and access scenarios in CGs corresponded to their requirements. Then, we identify the new features of users' behavior in Grid scheduling, such as dynamic, selfish, cooperative, trustful, symmetric and asymmetric natures. We also show how computational economy-based approaches, such as market mechanisms, and computational paradigms, such as Neural Networks, can be used to model user requirements and predict users' behaviors in CGs.

The rest of the paper is organized as follows. In Section II we define the main types of Grid users and provide a list of their requirements arising in Grid scenarios of data and resource management. Then, the case of users' relationship in scheduling processes in CGs is discussed and analyzed in Section III. The use of computational economy, game-theoretic and Neural Networks is presented in Section IV. We end the paper in Section V with some conclusions and indications for future work.

II. Users' layers in the grid architecture

In large-scale distributed Grid environments, Grid users can play many different roles: Grid administrator, community or Virtual Organization administrator, node administrator, service owner, user group administrator, service end-user, etc. Based on users' categorization defined in [Norman, 2006] we present below a summary of five main types of Grid users.

Grid-sys: *Infrastructure System Administrator*. This user is an expert in computing, his main duties are system administration of Grid nodes, possibly with infrastructure

978-0-7695-4237-9/10 $26.00 © 2010 IEEE
DOI 10.1109/3PGCIC.2010.94

delivery and security expertise. A Grid-Sys is likely to need to authenticate directly to particular grid resource nodes. This Grid-sys user is also in charge of monitoring the users activities and system performance. Advanced monitoring tools have been proposed in Grid computing domain for monitoring purposes.

SP: *Service Provider*. This user has expertise in computing, authorization and possibly identity management. The SP usually needs to authenticate, authorize and account for the Grid Service-End Users.

SEU: *Service End-User*. This user does not need to be an expert in computing, he also could not be familiar with the Grid architecture and management. His main interest is in submitting tasks and applications to be solved; additionally, he can upload the necessary data, scripts or source codes, can run queries, executable code or scripts via SPs. He can as well be defined as the user of applications served by SPs.

PUA: *Power User Agnostic* of grid resource node. The main activities of this user are the program development and data management. PUA usually is not concerned where processing takes place in the Grid system.

PUS: *Power User* requiring *Specific* Grid resource nodes. His duties and activities are similar to PUA's; additionally he may have more platform dependent and system administrative expertise. Also, Grid node owners may wish to have a direct authentication, authorization and accounting relationship with the PUS (differently from PUA users).

PUDS: *Power user Developing a Service* should have high degree of computing (like PUA and PUS) and developing expertise (like SP). If needed, he may interact with and accounting for SEUs in an experimental manner.

Grid user types arise and are located at different administrative domains, while operating at various levels of a wide Grid infrastructure. There is variety of ways, in which the access management requirements of each group of users and special polices defined by the resource owners, may be fulfilled. Each such scenario, as well as Grid technology in general, should evolve in response to the requirements of the users' communities. In Table I we present the users' general requirements with respect to the Grid administration, service, application and porting, usability and resource utilization (see also [EGI, 2009]).

It should be noted that the requirements on standardized authentication and authorization mechanisms as well as a globally accepted trustworthy of the Grid user are crucial at all Grid levels. The trustworthiness of the Grid user can be defined as a user authentication trustworthiness parameter (UAT), which can be expressed as a degree of user's trustworthiness who has passed system authentication, and must be taken as a basic qualification in judging access requests. The system should make an access control decision based on the users authentication trustworthiness. Although the user has passed system authentication, we can not make certain whether he is trusted or not. There are some uncertainties in authentication systems, such as the uncertainties of the authentication mechanisms, authentication rules and authentication conclusions. These uncertainties in the authentication process cane be modelled using the Fuzzy Logic. It has widely been applied in Grid computing for intrusion detection systems as well as for the prediction of the users' actions and decisions [Abraham et al., 2007], [Zhang et al., 2006]).

In other approaches the concept of authentication trustworthiness is analyzed and applied. In [Wang et al.,2004] the authentication trustworthiness Role-Based Access Control (RBAC) model was proposed. The model associates authentication trustworthiness RBAC model, and the authentication trustworthiness of the authenticated user is defined to activate his roles and permissions.

Another possible solution of the effective authentication/authorization could come directly from the Virtual Organizations (VOs). The users becoming members of a VO are authorized and have granted access to VO's resources. The VO membership contributes to distributing user management overheads and removes the need for replication of administrative efforts across the Grid. The use of VOs to administer users is an attractive feature of Grids. Thus the concept of VOs is also a key issue in the Grid architecture development and the designing a proper architectural model of Computational Grid (CG) is one of the most important issues in an efficient resources, tasks and users management according to the various users' requirements. The hierarchical layered structure seems to be well suited to capture the realistic administrative features of a real-life, large-scale distributed CG environment.

In the following sections we identity the main types of Grid users and their relationships in Grid scheduling.

III. Grid users relations in Grid scheduling

A. Hierarchical access model in Grid scheduling

The hierarchical Grid model can be interpreted as a result of hybridization of centralized and decentralized models [Xhafa et al., 2010]. The hierarchy in such a system can usually consists of two or three levels. The general concept of such systems is presented in Fig. 1.

The simplest example of the hierarchical bi-level Grid model is the Meta-Broker model (MB) [Garg et al., 2009]. In this system, Grid users submit their applications to the meta-broker. MB uses also the information supplied by the resource owners to match the users' tasks needs to appropriate machines.

The three-level Grid system is defined in Kwok et al. [Kwok et al., 2007]. The model is made up of the global, the inter-site and the intra-site levels. At the intra-site level, there is a federation of autonomous machines. The resource owners send an information about the computational capacities of the machines to the local managers, who defines the "Grid sites reputation indexes" and send them to the global

Table I
GRID USERS' GENERAL REQUIREMENTS

Requirement type	User type
Grid General Requirements	
Reliable grid middleware; Quality of Service	All users
Easy access to data and databases, fine grained access policies	All users
Advance reservation or the information when your job will be scheduled: monitoring of jobs, estimate of queue delay	All users
Standardized authentication and authorization mechanisms	All users
A globally accepted, trustworthy Grid user identity	All users
Administration Requirements	
Secure communication and transfer (e.g. TLS)	Grid-sys, PUDS, SEU
Uniform configuration across all Grid components	Gris-sys, PUDS
Ability to deploy and operate different versions of same software	PUA, PUS, Grid-sys
Standard mechanisms to fault tolerance, error handling, service recovery, outages and maintenance scheduling for all Grid components	Grid-sys,
Testing and monitoring of Grid components with automatic alerts	Grid-sys
Fast, easy site installation and verification procedures for new sites joining the grid	Grid-sys
Service Requirements	
Encryption and protection of data on grid storage elements	SP, PUA, PUS
Fast access and reliable transfer of massive amounts of data	SP, SEU, Grid-sys
Ad-hoc integration of arbitrary data sources that lie outside the Grid	SP
Application and Porting Requirements	
Consistent API for all middleware components	Gris-sys
Standardized error codes and error handling procedures	All users
Framework to easily use authentication, authorization and secure communication	SP, PUS, SEU
Client APIs for most prominent programming languages	PUA, PUS
Utilization and Usability Requirements	
Visualization of the completed results	SEU
Application steering modification of parameters during computation	SEU
Safe and easy authentication procedures	SEU, SP
Reliable real-time and instantaneous job submission for high priority jobs for e.g. risk and disaster management, recovery, etc.	SEU

scheduler. The global scheduler, operating at the global level, performs the tasks scheduling adaptively according to available scheduling algorithms implemented as well as scheduling modes (such as immediate and batch) specified by users for their tasks and applications. The roles of Meta-broker, Local Managers and Global Schedulers depend, in particular, on the End-Users' (SEUs') requirements defined in the following way:

Requirements for Specifying a Single Computational Task. At the basic level, the end-user will need to be able to specify and submit a single monolithic application (with well defined input and output data), a bag-of-tasks application with no dependencies among them. The user may as well be required by the system to provide information on types of tasks (e.g. data intensive *vs.* CPU intensive computing) and an estimation of task workload (e.g. in millions of instructions). In many cases users should be able to submit its tasks/applications as either executable or source code, in which case will need to be compiled and linked for further execution. The later, may require software deployment not available in the Grid node. Due to the complexity of the compilation and execution processes on heterogeneous architectures on the Grid, it is recommended to the users to build and test their application on a specific platform prior to submitting them to the Grid. Finally, in most cases data is assumed to be shipped with the task/application. In a more advanced setting, data should be accessed at some Grid nodes, in which case it is suggested to situate processing nodes "close" to data nodes.

Requirements for Specifying a Job of Multiple Tasks. The user should be able to specify a complex job that involves the execution of multiple tasks, the dependencies between which should be easily defined. The input and output data for the tasks must be defined for each task, providing the ability to specify relational data. Some graphical interfaces and representation of the graph structures (like Directed Acyclic Graph (DAG)) may be provided for the specification of tasks dependencies.

Access to Remote Data. The input and output data specified by the user may be stored remotely, thus he will need to provide the location of the remote data. If a ubiquitous wide-area file system is in operation on the Grid, the user would only have to care about the location of files and data with respect to some root location under which they are stored.

Resource Specification The user should be able to specify resource requirements, such as the number of processors and the hardware attributes required for that task. He may wish to target the particular types of resources (e.g. SMP machines), but, should not have to concern with the type of resource management on the Grid, nor with resource management systems on individual resources on the Grid.

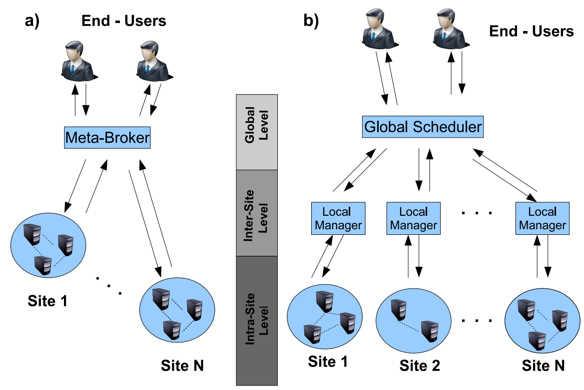

Figure 1. Two- and three-level hierarchical architectures of Computational Grids ((a) and (b), respectively).

Resource reliability. In some cases the machines in the grid system could be unavailable due to dynamics or special policies of the resource owners.The user should be able to get some information about the resource reliability to avoid or, at least, to reduce the cost of possible abortion of hist tasks. It is of course upon system's criteria to activate re-scheduling, task migration, and pre-emption policies.

Trustfulness of Resources - Secure Scheduling. The user could require to allocate his tasks in the most trustful resources. Thus he should be able to verify the trust indexes of the resources and define the security demands for his tasks on resources to be allocated to.

Standardized authentication and authorization mechanisms requirements. Users on the computational Grid will most likely make use of an standardized certificate authentication scheme. The certificates can be digitally signed by a certificate authority, and kept in the users repository, which is recognized by the resources and resource owners. It is desirable for a certificate to be automatically created by the user's interface application when a user submits a task to the Grid.

Job Monitoring and Control. Users should be able to monitor and control the progress status of their tasks.

B. *Grid users' relations in the scheduling process*

Several user behaviors and relationships scenarios that reflect the realistic access management models can be considered. We highlight below three basic relation schemes in Grid scheduling processes.

- **Non-cooperativeness:** In this scenario the users act independently of each other. It can be observed in a realistic Grid that the tasks/application submission is usually in which cooperation is difficult to happen at large scale. Grid users usually independently of each other submit their tasks/applications to the Grid system. Also the resource owners act selfishly in order to maximize the resource utilization. Their principal interest is to execute tasks from local users rather than contributing to the execution of remote tasks.
- **Cooperativeness:** In this case the users can form a coalition to plan in advance their actions. This model is useful for the intra-site Grid negotiations, where the local job dispatchers can define the joint "execution capabilities" parameters for the clusters of the Grid sites and notify them to the global scheduler.
- **Semi-cooperativeness:** In this model each user can select a partner for the cooperation. This scenario is applied in the multi-round auction models to incorporate rescheduling mechanism. The selection of the partner for cooperation is a complex issue and has been addressed in many research work. For instance, one can select trusted partners using trustworthiness models.

The users can have different privileges to the resources, so two following scenarios can be analyzed:

- **Symmetric scenario**. In this case there are no special privileges in the resource usage for the Grid users.
- **Asymmetric scenario**. In this case there is a privileged user (Leader), who can have a full access to resources as opposed to the rest of users who can be granted only limited access to resources. The Leader could also be the owner of a large portion of the task pool, as it is reasonable to allocate first his tasks at best resources in the system.

IV. Users' behavior models in Grid scheduling

A. *Computational economy and game-based models*

Market-based approaches in Grid computing enable Grid resource owners, acting as sellers, to earn money by letting others (mainly Grid End-users, acting as buyers) to use their (idle) computational resources. The pricing of resources is driven by demand and supply. These models can be easily translated into the game-theoretical frameworks and are useful in Grid resource management as well as in defining users' decision strategies.

In the following we briefly characterize most popular economically- and game-based approaches for modelling users' relations and decisions in scheduling process.

Commodity Market model: This model is based on the Meta-broker architecture. It is assumed here that the service providers primarily charge the end user for the resources they consume and the pricing policies are based on the demand from the users and the supply of resources. The resource owners and service providers are selfish in this approach and the end-users may cooperate or not [Buyya, 2009].

Auctions: In this model there are two groups of participants: sellers (resource owners) and buyers (Grid end-users). The cooperation between users to form a coalition and win the auction is possible, but usually users behave selfishly. The auction mechanism can be defined in many ways (e.g. English, Dutch, First and Second Price auctions). All of them differ in terms of whether they are performed as open or closed auctions and the offer price for the highest bidder. The users' strategies in particular auctions are discussed e.g. in [Ghosh et al. 2004].

Bi-level synchronized auctions: The First Price bidding auction mechanism has been extended by Kwok et al. [Kwok et al., 2007] to define the resource management and global scheduling policy at the intra- and inter-site levels in the 3-levels hierarchical Grid structure. In the intra-site bidding each machine owner in the site, who acts selfishly, declares the "execution capability" of the resource. The local manager monitors these amounts and sends a single value to the global scheduler. In the inter-site bidding the global scheduler should allocate tasks according to the values sent by the local dispatchers. The authors proved that the cooperation of the players at both levels are the optimal strategies for both level-auctions. However, for the successful execution of all those strategies some synchronization mechanism must be introduced, which can make the whole system inefficient in a large-scale dynamic environment.

Bargaining models: In this model the resource brokers bargain with resource providers for lower access price and longer usage duration. The negotiation process is guided by the end-users requirements (e.g., deadline) and can be provided directly between buyers (End-users) and sellers (resource owners). The most recent study on the bargaining cooperative model application in optimizing the energy consumption in Grid is proposed in [Subrata et al., 2010].

Non-cooperative and asymmetric end-users games: Economically-based models are suitable to exploit the interaction of different scheduling layers. However, in many cases Grid users and resource owners are likely to behave in different manners and can specify additional security and resource reliability requirements. In such a case their behavior cannot be characterized using conventional techniques. Game-theoretical models are quite natural tools for addressing this problem. All scheduling criteria can be aggregated and defined as a joint users' cost or pay-off functions, which make these kinds of models very useful in the analysis of the various users' strategies in the resource allocation process. There are many game scenarios, which can be adapted for modelling the users' behavior. Some of them, like bargaining games, can be simply adapted for the formalization of the economical mechanisms highlighted above. Recently, two general game scenarios, namely *non-cooperative* and *asymmetric* games, which takes into account the realistic feature of CGs, have been proposed.

In the non-cooperative model, the users act independently of each other. They are selfish and cannot form any coalition. This game scenario is very useful for defining the aggregate scheduling criteria. In [Garg et al., 2009] the authors propose the Grid users' game, in which the task execution and resource utilization costs are defined as the bi-objective players' cost functions. The resource owners selfish behavior in the hierarchical structure was modelled as a non-cooperative game in [Song et al., 2006]. The authors considered the risky and insecure conditions in online scheduling in Grids caused by software vulnerability and distrusted security policy. Recently, in [Kołodziej et al., 2009], the non-cooperative Grid End-users' game model was introduced for addressing the requirements for security and resource reliability in Grid scheduling.

On the other hand, the asymmetric End-users' model has been defined using the Stackelberg game scenario in [Kołodziej et al., 2010]. In this case one player (user) is acting as a Leader with a privileged access to resources. He assigns his tasks first, and the rest of the users (Followers) react rationally to the Leader's move. The Followers do not cooperate with each other, but their decisions depend on the Leader's action. This model illustrates very well the real-life situation, where the roles of the users are in fact asymmetric with regard to their access rights and usage of resources. It must also be noted that in many economical models the sellers and buyers stand in an asymmetric position as well. Having a control on a large resource pools and maintaining the large fraction of the task batch for scheduling can be the reasons of having some privileges in the resource access (especially in the case of the tasks with critical deadlines and some special reliability and security requirements) or in the setting of the reasonable resource utilization pricing policies.

B. *Neural network-based model*

As it was shown above, the game-based models can capture many realistic scenarios of Grid users' relations and activities. However, some other proposals can be considered as promising tools for defining complex decision making processes of various Grid users. In [Shelestov et al., 2008] the authors introduced the users' decision model based on the artificial neural network paradigm. Their model consists of three main components: (a) online module dedicated to the prediction of the users' actions; (b) off-line model based on the analysis of statistical data acquired during users work; and (c) change detection module defined for the detection of trends and changes in users' activities. The users' decisions mechanism are based on the feed-forward neural networks trained by the back-propagation method. This method can be considered as an alternative to game-based ones for the characteristics of the dynamics of the Grid users' decision process in the online scheduling. The authors additionally proposed the off-line model, where another neural network is applied, for the detection of normal/abnormal users activities, by analyzing the statistical data accumulated during the users' actions. It seems that the proposed approach can be a good start for some more detailed analysis of the users decision processes, however the complexity of the model can be a main drawback for its successful application in real-life Grid scenarios.

V. Conclusions and Future Work

In this work we presented a comprehensive set of user types, requirements and behaviors in Computational Grids (CGs). We have specified a categorization of the Grid users types and have defined their relationships and access scenarios in CGs corresponding to their requirements. The new features of users' behavior in Grid scheduling, such as dynamic, selfish, cooperative, trustful, symmetric and asymmetric natures are identified and analyzed. We have also discussed how computational economy, game-theoretic approaches and computational paradigms such as market mechanisms, non-cooperative games and neural networks can be used to model and predict users' behaviors and relationships in CGs. We believe that this analysis sets the basis for application designers in CGs to correctly embed user requirements and behaviors into Grid system. In our future work will use the HyperSim-G Grid simulator [Xhafa et al., 2007], to experimentally evaluate the identified set of user requirements and behaviors and their implications on Grid system functioning and performance.

References

[Abraham et al., 2007] A. Abrahama, R. Jainb, J. Thomasc, and S.Y. Hana: "D-SCIDS: Distributed soft computing intrusion detection system", *Journal of Network and Computer Applications* 30(1), 2007, pp. 81–98.

[Buyya, 2009] Buyya, R., and Bubendorfer, K. (Eds.): *"Market Oriented Grid and Utility Computing"*, Wiley, 2009.

[EGI, 2009] http://knowledge.eu-egi.eu/knowledge/index.php

[Garg et al., 2009] Garg, S.K., Buyya, R., and Segel, H.J.: "Scheduling Parallel Aplications on Utility Grids: Time and Cost Trade-off Management", *In Proc. of the 32nd ACSC*, 91. Mans, B., Ed. ACS., 2009, pp. 139–147.

[Ghosh et al. 2004] Ghosh, P., Roy, N., Basu, K., and Das, S.K.: "A Game Theory based Pricing Strategy for Job Allocation in Mobile Grids", *Proc. of the 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS 04)*, Santa Fe, New Mexico, 2004.

[Kołodziej et al., 2009] Kołodziej, J., Xhafa, F., Kolanko, Ł.: "Hierarchic Genetic Scheduler of Independent Jobs in Computational Grid Environment", *Proc. of 23rd ECMS, Madrid, 9-12.06.2009, in J. Otamendi, A. Bargieła, J.L. Montes and L.M. Doncel Pedrera eds.*, IEEE Press, Dudweiler, Germany, 2009, pp. 108–115.

[Kołodziej et al., 2010] Kołodziej, J. and Xhafa, F.: "A Game-Theoretic and Hybrid Genetic meta-heuristic Model for Security-Assured Scheduling of Independent Jobs in Computational Grids", *Proc. of CISIS 2010, Cracow, 15-18.02.2010, in L. Barolli, F. Xhafa and S. Venticinque eds.*, IEEE Press, , USA, 2010, pp. 93–100.

[Kwok et al., 2007] Y.-K. Kwok, K. Hwang and Sh. Song: "Selfish Grids: Game-Theoretic Modeling and NAS/PSA Benchmark Evaluation", *IEEE Transactions on Parallel and Distributing Systems*, 18(5), 2007, pp. 1–16.

[Norman, 2006] M. Norman: "Types of grid users and the Customer-Service Provider relationship: a future picture of grid use", *Proc. of the UK e-Science All Hands Meeting, Nottingham, 18-21.09.2006*, J.Cox Eds., 2006, pp. 37–44.

[Shelestov et al., 2008] A. Shelestov, S. Skakun and O. Kussul: "Intelligent Model of user Behavior in Distributed Systems", *Int. Journal on "Information Theories and Applications"* 15, 2008, pp. 70–75.

[Song et al., 2006] S. Song, K. Hwang, and Y-K. Kwok: "Risk-resilient Heuristics and Genetic Algorithms for Security- Assured Grid Job Scheduling, *IEEE Transactions on Computers*, 55(6), 2006, pp. 703-719.

[Subrata et al., 2010] Subrata, R., Zomaya, A. Y., and Landfeldt, B.: "Cooperative power-aware scheduling in grid computing environments", *J. Parallel Distrib. Comput.*, 70, 2010, pp. 84–91.

[Xhafa et al., 2010] Xhafa, F., Abraham, A.: "Computational models and heuristic methods for Grid scheduling problems", *Future Generation Computer Systems*, 26, 2010, pp. 608–621.

[Xhafa et al., 2007] F. Xhafa, J. Carretero, L. Barolli and A. Durresi: "Requirements for an Event-Based Simulation Package for Grid Systems". *J. of Interconnection Networks*, World Sci. Pub., 8(2), 2007, pp. 163–178.

[Wang et al.,2004] L. Wang, L. Wei, X. Liao and H. Wang: "AT-RBAC: An Authentication Trustworthiness-Based RBAC Model', *Proc. of Grid and Cooperative Computing GCC 2004 Workshops*, LNCS (3252), Springer Vlg., 2004, pp. 343–350.

[Zhang et al., 2006] G. Zhang and J. Sun: "Grid intrusion detection based on soft computing by modeling real-user's normal behaviors", *Proc. of IEEE International Conference onGranular Computing 2006*, pp. 558–561.